



**UNIVERSIDADE FEDERAL DO CEARÁ**  
**CURSO DE ENGENHARIA DA COMPUTAÇÃO**

**DISCIPLINA: ESTRUTURA DE DADOS**  
**PROFESSOR: JARBAS JOACI SÁ DE MESQUITA JUNIOR**  
**ALANNA MARIA MACHADO ALVES PAIVA**  
**MATRÍCULA: 421942**

**TRABALHO 03**

ALANNA MARIA MACHADO ALVES PAIVA

MATRÍCULA: 421942

### **TRABALHO 03**

Trabalho apresentado como requisito avaliativo para a disciplina de Estrutura de Dados em 2021, pelo Curso de Engenharia da Computação da Universidade Federal do Ceará – UFC.

Sobral – CE

2021

# SUMÁRIO

1. Introdução
2. Tabelas e Gráficos
3. Observações
4. Conclusões

## 1.INTRODUÇÃO

Esse trabalho tem como objetivo a implementação dos algoritmos BubbleSort, InsertionSort, QuickSort, MergeSort e Heapsort, a fim de calcular o tempo médio de cada um para assim, ordenar vetores com valores aleatórios de tamanho  $10^n$ , seja eles com  $n = \{1, 2, 3, \dots, 6\}$ .

Bubble sort é o algoritmo mais simples, mas o menos eficiente.

InsertionSort é um algoritmo simples e eficiente quando aplicado em pequenas listas. Neste algoritmo a lista é percorrida da esquerda para a direita, à medida que avança vai deixando os elementos mais à esquerda ordenados.

QuickSort é o algoritmo mais eficiente em listas totalmente desordenadas, ele se torna muito eficiente em relação aos outros no quesito de tempo.

MergeSort é um exemplo de algoritmo de ordenação por comparação do tipo dividir-para-conquistar.

Heapsort é uma estrutura de dados, conhecida como heap, que enxerga o vetor como uma árvore binária.

## 1 ° TENTATIVA

Tabela dos dados gerados:

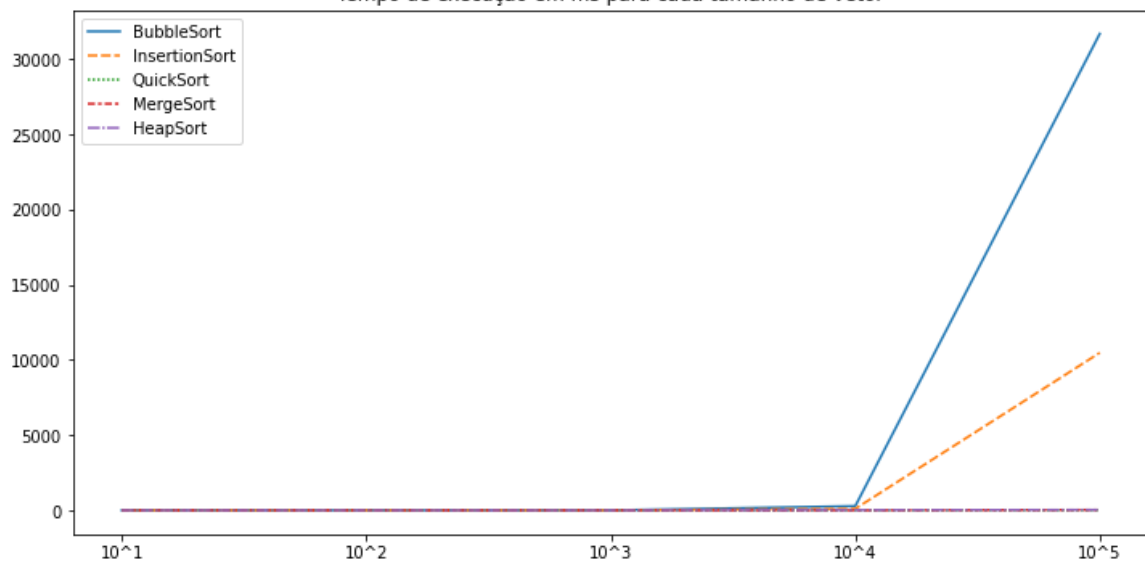
	BubbleSort	InsertionSort	QuickSort	MergeSort	HeapSort
<b>10<sup>1</sup></b>	0	0	0	0	0
<b>10<sup>2</sup></b>	0	0	0	0	0
<b>10<sup>3</sup></b>	2	1	0	0	1
<b>10<sup>4</sup></b>	286	97	0	0	16
<b>10<sup>5</sup></b>	33018	10485	16	38	29

Gráfico:

Figura 1: tamanho do vetor em função do tempo

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f45d402f750>

Tempo de execução em ms para cada tamanho de vetor



## 2º TENTATIVA

Tabela dos dados gerados:

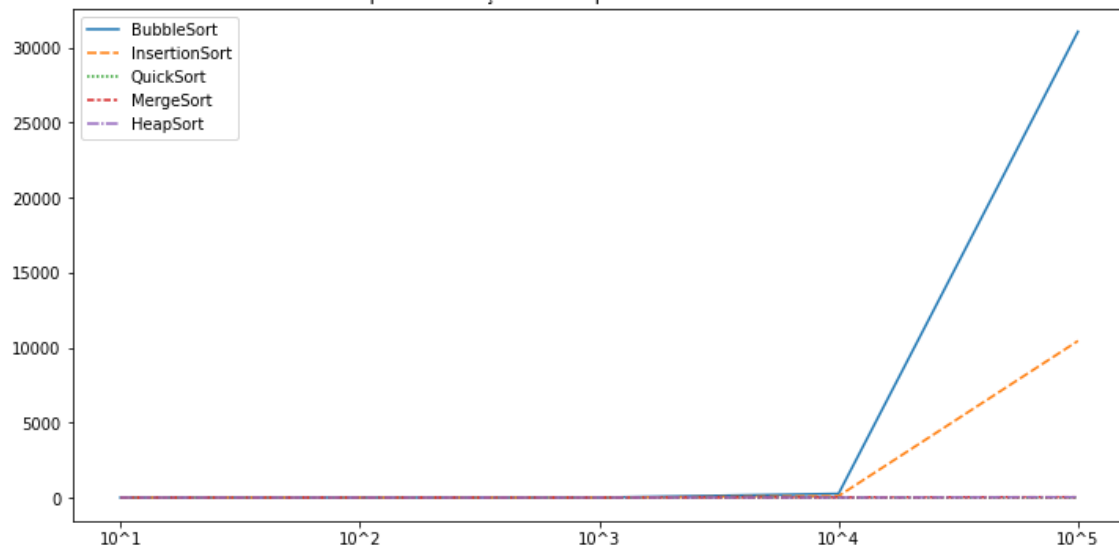
	BubbleSort	InsertionSort	QuickSort	MergeSort	HeapSort
<b>10<sup>1</sup></b>	0	0	0	0	0
<b>10<sup>2</sup></b>	0	0	0	0	0
<b>10<sup>3</sup></b>	0	0	0	0	0
<b>10<sup>4</sup></b>	254	97	0	16	0
<b>10<sup>5</sup></b>	31075	10445	0	31	24

Gráfico:

Figura 2: tamanho do vetor em função do tempo

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f45e2f63e50>

Tempo de execução em ms para cada tamanho de vetor



### 3° TENTATIVA

Tabela dos dados gerados:

	BubbleSort	InsertionSort	QuickSort	MergeSort	HeapSort
10 <sup>1</sup>	0	0	0	0	0
10 <sup>2</sup>	0	0	0	0	0
10 <sup>3</sup>	0	0	0	0	0
10 <sup>4</sup>	281	101	0	0	15
10 <sup>5</sup>	31706	10481	11	30	30

Gráfico:

Figura 3: tamanho do vetor em função do tempo

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f45d402f750>



## 4° TENTATIVA

Tabela dos dados gerados:

	BubbleSort	InsertionSort	QuickSort	MergeSort	HeapSort
$10^1$	0	0	0	0	0
$10^2$	0	0	0	0	0
$10^3$	0	0	0	0	0
$10^4$	256	94	6	3	1
$10^5$	31423	10378	17	31	31

Gráfico:

Figura 4: tamanho do vetor em função do tempo

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f45d3ebfb50>





## 5° TENTATIVA

Tabela dos Dados gerados:

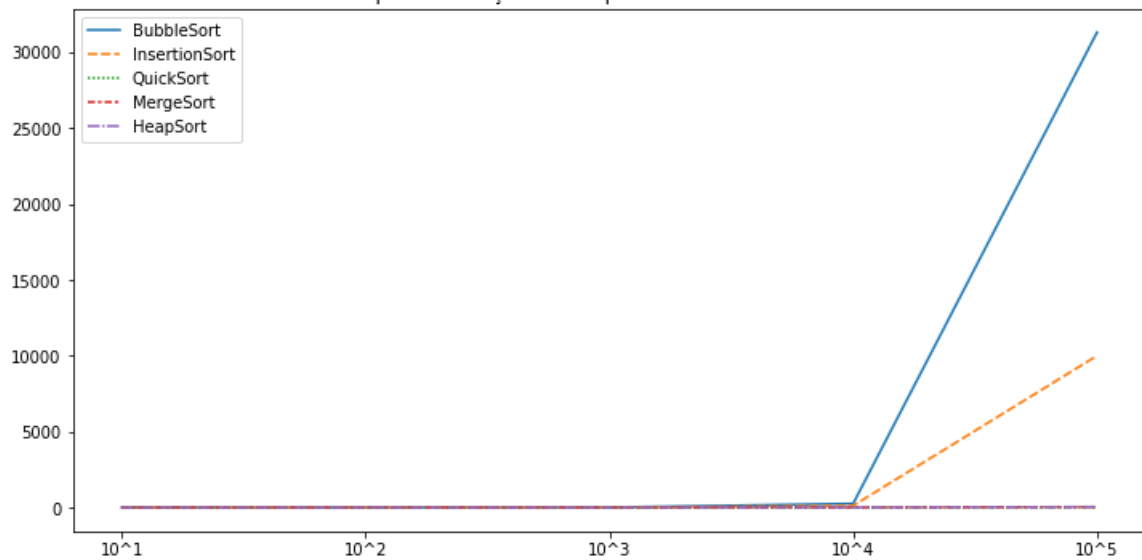
	BubbleSort	InsertionSort	QuickSort	MergeSort	HeapSort
<b>10<sup>1</sup></b>	0	0	0	0	0
<b>10<sup>2</sup></b>	0	0	0	0	0
<b>10<sup>3</sup></b>	0	0	0	0	0
<b>10<sup>4</sup></b>	250	110	0	0	0
<b>10<sup>5</sup></b>	31316	10000	16	31	32

Gráfico:

Figura 5: tamanho do vetor em função do tempo

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f45d3e342d0>

Tempo de execução em ms para cada tamanho de vetor



## **OBSERVAÇÕES E CONCLUSÕES**

Após a realização dos testes, foi possível observar que houve uma exorbitante variação entre eles, por isso, deve-se escolher aquele que apresenta um melhor aproveitamento de tempo.

Os gráficos foram gerados utilizando a biblioteca Matplotlib em Python, foram feitos vários testes para conferir os resultados, os números são aproximados e por isso, não é possível notar grande diferença entre os gráficos de cada tentativa.

Além disso, foi calculado o tamanho dos vetores até  $10^5$ , pois para  $n$  igual a 6 houve problema. Ademais, outros alunos tiveram problemas sérios com os seus computadores, logo, não quis tentar novamente e pôr em risco meu único computador.